Open NFC - Core Edition - Release Notes

Document Type:	Release Notes
Reference:	REN_NFC_1202-303 Version 0.5 (14719)
Release Date:	Sept. 14, 2012
File Name:	REN_NFC_1202-303 Open NFC - Core Edition - Release Notes.pdf
Security Level:	General Business Use

Copyright © 2012 Inside Secure

Disclaimer

This document is licensed under the Creative Commons Attribution 3.0 license (http://creativecommons.org/licenses/by/3.0/). (You may use the content of this document in any way that is consistent with this license and if you give proper attribution (http://www.open-nfc.org/license.html#attribution).

Copyright © 2012 Inside Secure

Open NFC and the Open NFC logo are trademarks or registered trademarks of Inside Secure.

Other brand, product and company names mentioned herein may be trademarks, registered trademarks or trade names of their respective owners.

History

Version	Date	Comments
0.1	Feb. 15, 2012	Release for Open NFC 4.4.0
0.2	March 9, 2012	Release for Open NFC 4.4.1
0.3	April 20, 2012	Release for Open NFC 4.4.2
0.4	June 29, 2012	Release for Open NFC 4.4.3
0.5	Sept. 14, 2012	Release for Open NFC 4.5.0

Copyright © 2012 Inside Secure

Summary of Contents

1 Introduction 5

2 Contents 6

3 Win32 Porting 8

4 Change Log 9

4.1 Changes between version 4.5.0 and version 4.4.3 9
4.2 Changes between version 4.4.3 and version 4.4.2 9
4.3 Changes between version 4.4.2 and version 4.4.1 10
4.4 Changes between version 4.4.1 and version 4.4.0 12
4.5 Changes between version 4.4.0 and version 4.3.3 13

Copyright © 2012 Inside Secure

1 Introduction

This document is the release notes of the NFC software stack “Open NFC - Core Edition v4.5.0”.

The C API of version 4.5.x is almost backward compatible with the previous versions 4.x of the stack. The API changes are described in the detailed list of the issues/changes done since the previous release.

The C APIs of the next versions 4.x of the product will remain backward compatible with this version of the API.

The HAL APIs of the product are stable enough to undertake a porting effort and to expect a minimum work with the next versions 4.x of the product.

The changes in this version compared to the previous version are listed in the last section of the document.

The following tools are validated to recompile the Windows porting of Open NFC and the examples of applications:

- “Visual C++ 2010 Express Edition” The Express edition is a free version of Visual C++ 2010. This tool may by download from Microsoft MSDN web site.
- “Windows SDK” The Windows SDK contains the files and the tools to compile Win32 applications. The Windows SDK can be downloaded from the following URL: http://www.microsoft.com/msdownload/ the Windows SDK used to validate the Win32 porting is "Windows SDK 7.1".

These tools are optional. Other Win32 compilation tool chains may be used to recompile the Windows projects but they are not validated for this delivery.

Copyright © 2012 Inside Secure

2 Contents

This delivery contains the following elements:

- the implementation of Open NFC - Core Edition in portable C source code,
- an example of porting on windows 7,
- the NFC Simulator tool for PC
- some examples of applications and
- the product documentation for the client API and the porting of the stack.

The delivery contains the following files and folders:

./

- REN_NFC_1202-303 Open NFC - Core Edition - Release Notes.pdf
  The Release Notes of this product (this file).

./sources:

- The C source code of the implementation.

./sources/autogen:

- The tool used to generate some C source code of the implementation.

./interfaces:

- The C header files containing the API for the porting HAL and the C API.

./porting:

- *SIS_NFC_0709-014 Open NFC Core Edition - Porting Guide.chm*
  The Porting Guide document describes the porting HAL and the porting process.

- *SIS_NFC_1005-194 Open NFC - NFC HAL API Specification.chm*
  This document describes the specification of the interface between the NFC software stack and the NFC Controller.

- *SIS_NFC_0806-058 Open NFC - NFC HAL Protocol Specification.pdf*
  This document describes the protocol used for the NFC HAL.

./porting/win32:
./porting/win32x64_common:
./porting/ccclient:

- An example of porting for Windows 7 32/64-bit. See section below.

./porting/win32/win32_examples:

- Some examples of applications to test Open NFC - Core Edition under Windows 7.

  The document “MAN_NFC_0901-099 Open NFC - PC Edition - Examples - User's Manual.pdf” describes how to build and how to use the applications.

./porting/jni:

- The Java wrapper for the NFC API on top of Open NFC. This directory includes the Open NFC JAR file, the Java API documentation, the source of the implementation and some examples of Java applications in source code.
- *SIS_NFC_1003-188 Open NFC - Java Developer's Manual.zip*
  The Java API documentation

./security

Copyright © 2012 Inside Secure

- *FRS_NFC_1104-241 Open NFC - Security Stack.pdf*
- *STS_NFC_1104-242 Open NFC - Security Stack Specification.pdf*
  The security stack specifications

./security/aclgen

- ACL Generator Tool, a PC tool designed to generate the ACL binary stream to be written into the EF(SE-ACF) file of "PKCS#15 Application

- *FRS_NFC_1104-244 Open NFC - Security Stack - AC File Generator Tool.pdf*
  The user manual for the ACL Generator tool.

/nfcc_simulator:

- *NFCSimulator.exe*
  The executable of the simulator of NFC Controller on PC.

- *MAN_NFC_0905-114 Open NFC - NFC Controller Simulator - User's Manual.pdf*
  The manual describing the usage of the NFC Simulator.

/nfcc_simulator/virtualcards:

- The files used for the simulation of the cards.

./connection_center

- *Connection Center.exe*
  The connection center executable and the corresponding DLLs.
- *MAN_NFC_0904-106 Open NFC - Connection Center - User's Manual.pdf*
  The user's manual for the Connection Center tool.

./documents:

- *SIS_NFC_0707-003 Open NFC - API Specification.chm*
  The C API documentation for the applications.

- *STS_NFC_0707-001 NFC Tag Type 5 Specification.pdf*
  Specification for the Type 5 Tags

- *STS_NFC_0707-002 NFC Tag Type 6 Specification.pdf*
  Specification for the Type 6 Tags

- *PRE_NFC_0804-250 NFC Standards.pdf*
  An overview of the NFC standards and products

Copyright © 2012 Inside Secure

3 Win32 Porting

The Win32 porting is an example of porting for testing the code on PC under Windows XP/7. This part is an unsupported delivery. The win32 porting folder contains a MS Visual C++ 2010 project and the source code of the Win32 porting.

Compilation

To recompile the project, you must use MS Visual C++ 2010 Professional Edition, or the free "Express Edition".

The product "MS Visual C++ 2010 Express Edition" is downloadable from Microsoft web site.

Porting Specificities

The Win32 porting is just an example because it is not designed as a real porting should be. The main differences are the following:

- The Win32 porting is a console application, while the stack should be ported in a driver.
- The timer of the NFC Controller communication stack uses timeout values as little as 10ms. This value is required for an optimal usage of the NFC Controller. The accuracy of the Win32 scheduling for the applications is closer to 40ms.

Porting Execution

Use the examples provided with the delivery.

Copyright © 2012 Inside Secure

4 Change Log

This section describes differences between the current release and the previous versions of the product.

4.1 Changes between version 4.5.0 and version 4.4.3

Id.	Type	Description
WME-1715	New feature	Added minimal support of the Secure Element VaultSEcure v1.0 on DCLB (chip reference AT90SC352208RCV).
WME-1718	New feature	Added new API WNDEFGetTagInfoEx(). This API provides all the information provided by WNDEFGetTagInfo(), plus the size of the tag. The API WNDEFGetTagInfo() is deprecated.
WME-1734	New feature	Added new WDTACompatibilityEnable () and WDTACompatibilityDisable() APIs. These API allow to slightly modify the behaviour of the stack to allow to pass DTA campaign.
WME-1735	New feature	Added API to allow raw NFC DEP exchanges (e.g NFC DEP data exchanges that do not carry LLCP payload). See client API for details
WME-1763	Change	Improvement of end of transaction event accuracy in case of communication errors with the SE.
WME-1727	Change	Several changes have been introduced in the management of the end of transaction events. The interrogation of the embedded SE when an end of transaction event is received is now optional. It is enabled by default (backward compatible), and can be disabled by defining P_EXCLUDE_GET_TRANSACTION_AID in the porting_config.h. When enabled, the interrogation of the embedded SE is only done when user registered a transaction event listener (previous behaviour was to access to the SE regardless of the presence of an event listener)
WME-1732	Bug fix	In client/server porting, the server stops itself if the client calls a driver api requiring a buffer (ex PEmulGetMessageData) with a null value length.
WME-1729	Bug fix	Cannot detect Mifare ULC card which has access right preventing access to the NDEF data area: A NACK is returned by the Mifare ULC Card during the identification of the NDEF Type 2. The check of the Capability Container has been added before getting the data contained in the message.
WME-1746	Bug fix	Fix null pointer dereference which would occur if CMemoryAlloc failed in PNDEFRegisterSNEPMessageHandlerDriver()
WME-1748	Bug fix	Fix null pointer dereference upon call of W7816GetAid with pnActualLength == NULL.
WME-1743	Bug fix	Buffer overflow in P14P3DriverExchangeRawBits when sending payload of 253 or 254 bytes. The data length verification has been modified to check the payload length is not greater than 252 bytes
WME-1744	Bug fix	Buffer overflow in static_P7816StoreAtr when reading 14443-4A card which presents ApplicationDataLength > 43 The length of ApplicationData is truncated.
WME-1747	Bug fix	Memory used after CMemoryFree in static_PKovioRFIDCreateConnectionAutomaton() on fail
WME-1745	Bug fix	Null pointer dereference upon CMemoryAlloc fail in PHandleDecrementReferenceCount(). Removed the call of static_PHandleDestroy in case of error during the memory allocation.

4.2 Changes between version 4.4.3 and version 4.4.2

Id.	Type	Description
WME-1700	New feature	Routing table support
WME-1672	New feature	Modification of the reader registry introducing the P_READER_DETECTION_RETRIES which defines the number of detection of the card if an error occured. If an RF error occurred during the detection the reader registry retries the same detection again until the number of detection equals the P_READER_DETECTION_RETRIES value.

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1700	New feature	Routing table support
WME-1672	New feature	Modification of the reader registry introducing the P_READER_DETECTION_RETRIES which defines the number of detection of the card if an error occured. If an RF error occurred during the detection the reader registry retries the same detection again until the number of detection equals the P_READER_DETECTION_RETRIES value.
WME-1639	Change backward compatible	Add a trace message with the stack version at boot
WME-1702	Change backward compatible	Do no longer use deprecated APIs in the NDEF Type 4 implementation
WME-1703	Change backward compatible	Cleaning-up and factorizing the ISO 7816 constants
WME-1704	Change backward compatible	Secure element code cleanup
WME-1625	Bug fix	During the parsing of the block security status, one byte is read after the end of the answer, leading to possible crash / invalid lock configuration.
WME-1626	Bug fix	Infinite loop in the stack when a ISO15693 card with 256 blocks is detected
WME-1623	Bug fix	An Operation handle is closed twice in case of error during SNEP or NPP exchange.
WME-1624	Bug fix	Memory leak during card removal detection for Type1 chip
WME-1628	Bug fix	Calling WSESetPolicy during a RF transaction with the embedded SE returns W_ERROR_BAD_STATE and the SE Policy becomes inconsistent
WME-1634	Bug fix	if WSESetPolicy is called during a transaction, the EndOfTransaction event is not monitored
WME-1635	Bug fix	In P2P connectionless mode, received packets are lost when enqueuing more than two packets
WME-1638	Bug fix	the SNEP Server does not restart the P2P after receiving a first NDEF Message
		Add the support of STCK_ACTIVATE, STCK_REFRESH and HOTPLUG in the secure element porting
WME-1645	Bug fix	WSEGetInfoEx returns property W_NFCC_PROTOCOL_CARD_MIFARE_PLUS for UICC
WME-1636	Bug fix	WEmul API returns W_ERROR_BAD_HANDLE instead of W_ERROR_CONNECTION_COMPATIBILITY
WME-1648	Bug fix	The function WReaderExchangeData does not work with Kovio RFID Cards
WME-1649	Bug fix	WP2PShutdown callback is not called when WP2PShutdown with WP2PConnect is pending.
WME-1651	Bug fix	WP2PConnect returned W_SUCCESS instead of W_ERROR_BAD_STATE when the socket is already connected
WME-1673	Bug fix	If a PKCS#15 applet returns only 9000 (without FCP data) when it is selected, it is not managed by the Security Stack even if it is compliant with GSMA specifications. Then, access to the SIM is denied.
WME-1674	Bug fix	After an End Of Transaction Event with a UICC, Access Control Rules cannot be read if a user has already opened a connection with this UICC.
WME-1681	Bug fix	In P2P target mode, the response must be sent before expiration of RWT and not LTO
WME-1687	Bug fix	Prevent use of variable shadowing : checked using the mingw32 port. Fixes a bug in static_PSEBuildInfo, variable shadowing was not a problem in other places of the code.
WME-1644	Bug fix	WEmulSendAnswer returns W_SUCCESS instead of BAD_PARAMETER on User Driver Mode only
WME-1695	Bug fix	Sometimes the NAL returns a wrong kovio identifier after detecting a KOVIO card. After unscrambling the data, the CRC check is done. In case of mismatching, the polling is restarted.
WME-1701	Bug fix	After a call of a Jupiter Operation (Get/Set Policy, Get Transaction AID), if an error occured in PSEDriverOperation the memory allocated for the jupiter instance struct is lost

4.3 Changes between version 4.4.2 and version 4.4.1

Id.	Type	Description
WME-1601 WME-1581	New feature	Add support of Kovio RFID Type 2 tags The support includes format, write access for initialized (NDEF TLV length = 0) tags and read only access for Tags that already contain a NDEF message (NDEF TLV length > 0). A new connection property W_PROP_KOVIO_RFID is defined.

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1577 WME-1536	New feature	Reading a NDEF message containing chunk records is now possible. However, the modification of the message content is not allowed.
WME-1578	Change backward compatible	Remove obsolete, no longer used partial reading of NDEF messages from the source code. The partial reading of the NDEF messages from the tag has been removed for a while since it does not allow properly checking of the correctness of NDEF messages. The corresponding dead code has been removed.
WME-1583	Change backward compatible	Improve reader registry to ensure all supported technologies will be detected when presenting a multi-protocol device.
WME-1582	Change backward compatible	Improved compliance to NFC Forum for Tag type 1, 2, 3, 4
WME-1606	Bug fix	Bug in the implementation of WSECheckAidAccess() The implementation returns a W_ERROR_SECURITY if the caller is not authenticated. Authentication may not be required in some porting because WSECheckAidAccess() uses impersonation. The optional requirement for authentication of the caller should checked by the porting in the implementation of CSecurityCheckImpersonatedIdentity()
WME-1607	Bug fix	Removing the filtering of the AID in the transaction event. This shall be done in the event dispatcher of each specific porting.
WME-1584	Bug fix	two warning occur during the compilation of the smart cache file: - The first is an unused variable in release mode but used in debug (Assert) - The second is the possibility of access to out of bound array index
WME-1587	Bug fix	W7816OpenLogicalChannelSync() accepts nAIDLength = 0 while pAID != null
WME-1590	Bug fix	The aREPGEN array of the tWBPrimeConnectionInfo is too short to contain whole REPGEN content
WME-1574	Bug fix	Fixes several possible issues detected by Klockwork (possible memory leak in error cases)
WME-1585	Bug fix	W7816GetATRSize() return 0 for handle retrieved by WSEOpenConnectionSync()
WME-1586	Bug fix	W7816GetResponseAPDUData() does not provide APDU after calling W7816ExchangeAPDUSync() with too short buffer
WME-1592	Bug fix	During processing of the GlobalPlatform ACL, the security stack refuses to process the EF(ODF) and EF(ACRF) files if they are larger than 256 bytes, which aborts the ACL reading.
WME-1597	Bug fix	Card Removal Detection must return the event "card removed" when the hal returns RF Error (timeout, RF Protocols, ...)
WME-1579	Bug fix	When DSFID or AFI is not supported (zero value) , DSFIDLocked or AFILocked should be TRUE.
WME-1507	Bug fix	When a collision between ISO14443A, ISO14443B and Type 1 is generated by the Simulator, the collision event handler is not called The stack no longer assumes that NAL layer will perform proper technology grouping when a collision is detected.
WME-1604	Bug fix	Some internal handles are not closed when erroneous use of data exchange functions. Impacts Mifare, Type 1, Picopass, NDEF and ISO7816 API.
WME-1605	Bug fix	Heap corruption in static_PRTDURIGetValue() during WRTDURIGetValue() when invalid UTF8 characters in record

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1610	Bug fix	Possible memory leak in PLLCPProcessIncomingPDU
WME-1612	Bug fix	WNFCControllerGetProperty function returns W_SUCCESS while expected W_ERROR_BUFFER_TOO_SHORT
WME-1615	Bug fix	Problem with the management of the embedded secure element transaction history for iClass transactions When an iClass end of transaction event was received, the transaction history was not cleared thus the transaction history was read on the next EOT event.
WME-1616	Bug fix	End of transaction event is not monitored if the default applet has been used When the NFCC spied a Select of the default applet (null AID), an end of transaction event was received with AID list = '00' and was ignored.
WME-1617	Bug fix	When an end of transaction event is received for the embedded SE, ACL is read twice
WME-1613	Bug fix	Read memory out of bounds in PRTDTextGetValue() with badly formated record
WME-1619	Bug fix	WNDEFCreateRecord and WNDEFCreateNestedMessageRecord allowed to create records using type W_NDEF_TNF_UNCHANGED, resulting in building an invalid NDEF message.
WME-1622	Bug fix	The function WP2PRecvFrom returns W_ERROR_BUFFER_TOO_SHORT if the given buffer length equals the received data length.

4.4 Changes between version 4.4.1 and version 4.4.0

Id.	Type	Description
WME-1454	New feature	Adding configuration variables to exclude the implementation of unwanted functionalities. The following configuration variables are defined: - P_EXCLUDE_DEPRECATED_FUNCTIONS excludes the implementation of deprecated functions. - P_INCLUDE_SE_SECURITY includes the Security Stack and the SE HAL (was already defined in previous versions) - P_INCLUDE_SNEP_NPP includes the SNEP and NPP. - P_INCLUDE_PAIRING includes the BT and Wi-Fi pairing - P_INCLUDE_TEST_ENGINE includes the Test Engine - P_INCLUDE_PICOPASS includes the support of the Picopass tags and the Type 5 tags - P_INCLUDE_MIFARE_CLASSIC includes the support of the Mifare Classic tags and the Type 7 tags
WME-1126 WME-1572 WME-1568 WME-1561 WME-1559	New feature	Support of the SNEP & NPP protocols to exchange NDEF message on top of P2P Adding the function WNDEFRegisterMessageHandler(), WNDEFHandlerWorkPerformed() and WNDEFSendMessage(). Adding the protocol constant W_PROP_NPP and W_PROP_SNEP. WNDEFReadMessageOnAnyTag() is deprecated.
WME-1558	New feature	Add NDEF Type 4 cache support Add NDEF Type 4 cache support to avoid entire reading the tag contents each time it is detected.
WME-1544	Bug fix	NFCControllerReset does not work in RAW mode NFCControllerReset does not work in RAW mode. The Server must be closed and restarted after a client switched to RAW mode.
WME-1546	Bug fix	A CCClient MSG_CONNECT_ERR_NO_PROVIDER error is displayed in the console when starting Open NFC without pcsc_server When the pcsc_server (iso7816_4 service provider) is not running, CCClient displays a MSG_CONNECT_ERR_NO_PROVIDER error in the console at the begining of the Open NFC's boot.

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1517	Bug fix	WP2PConnect returned W_SUCCESS instead of W_ERROR_BAD_STATE when socket is already connected.
WME-1547	Bug fix	URI lookup callbacks and Read/Write callbacks are not called when the RF link is broken
WME-1549	Bug fix	WRTDTextGetValue returns inconsistent endianness string While most of the functions use UTF16 encoded according host endianess in the stack, the buffer returned by WRTDTextGetValue is UTF16BE in case the tag contains UTF16 BE, and UTF16 if the tag contains UTF8. The memcpy operation at the end of PRTDTextGetValue should be changed to a smarter operation (depending on system's UTF16 format).
WME-1550	Bug fix	In win32 secure element porting, the callback function of the SE HAL is called directly from the function implementation
WME-1526	Bug fix	Open connection with null card emulation parameter structure failed in client server mode only
WME-1551	Bug fix	WEmulOpenConnection returns W_ERROR_FEATURE_NOT_SUPPORTED as expected but does not set phandle to null.
WME-1560	Bug fix	Avoid crash when detecting an badly formatted NDEF type 4 tag
WME-1537	Bug fix	Card Removal Detection does not work with Gemalto UICC mNFC1.2
WME-1562	Bug fix	Possible loss of LLCP PDUs when a connection or disconnection collision occurs
WME-1565	Bug fix	Error W_ERROR_NFC_HAL_COMMUNICATION appears when restoring of the 14443-4A default reader parameter
WME-1566	Bug fix	Crash when writing in a Micropass NDEF Type 4
WME-1567	Bug fix	When processing P2P link activation, depending on the parameters sent by the peer device, the stacks uses unitialized value
WME-1575	Bug fix	WNDEFWriteMessageOnAnyTag does not work on formatted Mifare UL-C tags with authentication threshold set to 0x28

4.5 Changes between version 4.4.0 and version 4.3.3

Backward compatibility of the new ISO 7816 API
If the variable P_EXCLUDE_DEPRECATED_FUNCTIONS is not defined, the new API for ISO 7816 is backward compatible with the previous version. However, the following limitations should be considered:

The connection handle may be used with the function W7816ExchangeAPDU() function to be backward compliant with the previous version of the API. In this case, the connection handle is implicitly considered as a raw channel. This implies the following limitations:

- The function W7816ExchangeAPDU() cannot be used with the connection handle if a channel is already open (raw, basic or logic channel). The call will fail with the error W_ERROR_EXCLUSIVE_REJECTED.

Copyright © 2012 Inside Secure

- The function W7816OpenLogicalChannel() cannot be used if the connection handle has been used once with the function W7816ExchangeAPDU(). The call will fail with the error W_ERROR_EXCLUSIVE_REJECTED.
- The function W7816ExchangeAPDU() used with the connection handle does not filter the CLA byte of the APDU.

Id.	Type	Description
WME-1477	New feature	Adding the Mifare Classic HAL. This HAL is optional. The implementation is required only if the new configuration variable P_INCLUDE_MIFARE_CLASSIC is defined.
WME-1476	New feature	Adding the support of the bit-oriented mode for the reader ISO 14443-3 A - Adding the functions W14Part3ExchangeRawBits() and W14Part3ExchangeRawBitsSync(). - Adding the new NFC Controller capability W_NFCC_PROP_READER_ISO_14443_A_BIT
WME-1465	New feature	Updating the SE HAL to support SIM Alliance, GSMA and OGDR specifications Adding the support of the TCK command REFRESH in the SE HAL: The UICC sends this command to notify the device of a change in the UICC status. The security stack uses this notification to update the content of the policy files. See section 6.6.13 of the specification ETSI TS 102 223 V8.0.0 (2008-07) "Card Application Toolkit (CAT) (Release 8)" to get further information on the REFRESH command. The following changes are made in the SE HAL: - Adding a buffer receiving the identifiers of the modified files in the signature of the function CSeCreate() - Adding the suppport for the STK command REFRESH with the event C_SE_NOTIFY_STK_REFRESH. - Renaming the constant C_SE_OPERATION_HOT_PLUG into C_SE_NOTIFY_HOT_PLUG. - Renaming the constant C_SE_OPERATION_ACTIVATE_SWP into C_SE_NOTIFY_STK_ACTIVATE_SWP. - Adding the flag C_SE_FLAG_STK_REFRESH_SUPPORT. - Adding a SWP timout parameter in the signature of the function CSeGetStaticInfo(). - Renaming CSeOpenLogicalChannel() into CSeOpenChannel(). - Adding the possibility to open a raw channel or a basic channel and to select the master file (MF). - Renaming CSeCloseLogicalChannel() into CSeCloseChannel(). - Adding the function CSeGetResponseApdu(). The changes are required to implement the following specifications: - SIM Alliance v2.02 - GSMA NFC Handset APIs & Requirements - Version 2.0 - November 2011 - OGDR v11.
WME-1457	New feature	Implementation of the SIM Alliance API version 2.02 - Transport API
WME-1456	New Feature	Added ISO 7816 API to manage the raw APDU and the basic channel. Adding the functions W7816OpenRawChannel(), W7816OpenBasicChannel(), W7816OpenBasicChannelSync() and W7816GetAid(). For backward compatibility of the API, see the release notes.
WME-1330	New feature	Adding new functions WReaderExchangeDataEx() and WReaderExchangeDataExSync() to read a tag with a specific protocol
WME-1315	New feature	The Security Stack now support the existing PKCS#15 applet for JSR 177-like security and the security defined in the GSMA specification: NFC Handset APIs & Requirements - Version 2.0 - November 2011 GSM Association - http://www.gsm.org If the Secure Element is a UICC and there is no PKCS#15 compliant applet or file system in the UICC, the behaviour of the Security Stack is defined by the variable P_NO_UICC_ACCESS_BY_DEFAULT. If this variable is defined, the Security Stack rejects any access to the UICC. If the variable is not defined, the Security Stack allows every access to the UICC (except on the basic channel).
WME-1114	New feature	Support for the Mifare Classic reader functions including the NFC Tag on Mifare Classic

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-647	New feature	Add utility functions in the C API to display the string value the constants (protocol, properties,...) Adding the functions: WBasicGetConnectionPropertyName() and WBasicGetErrorString().
WME-1531	Change	Adding the length of the authentication data and of the user data in the Security HAL A length parameter is added in the Security HAL functions Adding the function CSecurityGetIdentityData().
WME-1480	Change backward compatible	Remove 7816 Support in CC and Clean up documentation (Backward compatible since the feature was not used at all)
WME-1427	Change backward compatible	Renamed internal error constant W_ERROR_POLLING_PENDING into W_ERROR_OPERATION_PENDING
WME-1321	Change backward compatible	Redevelop the ISO 7816 layer as a state machine: clean up all layers Isolate the 7816 function as a separate state machine accessed through interfaces Use at different places in the implementation Remove the need for fragmentation-reassembly of the ISO 7816 protocol: Remove a lot of code and errors
WME-1571	Change backward compatible	Improve robustness by restarting automatically the detection if an error occurs during the card found processing
WME-1432	Bug fix	WReaderExchangeDataSync leads the display of an error message on the client side Each time the function WReaderExchangeData|WReaderExchangeDataSync is called, this error is displayed: ERROR HANDL PHandleGetObject: Illegal value for the handle
WME-1442	Bug fix	W15Read() returns W_ERROR_BAD_HANDLE while expected W_ERROR_CONNECTION_COMPATIBILITY with incompatible tags.
WME-1439	Bug fix	Fixed WReaderListenToCardRemovalDetection for Felica Lite card
WME-1443	Bug fix	Infineon SLE 66R232S are not detected by Open NFC When trying to detect an Infineon SLE 66R32S card, no card handler is called. Should be detected as a A3 card
WME-1445	Bug fix	Programmation of TagIt CID / DSFID randomly fails Depending on the timing of the stack, the programmation of CID / DSFID on Tag IT succeeds or fails. The problem occurs when the SOF sent to the tag is issued too fast after the write command (the Tag did not have the time to complete the write operation)
WME-1447	Bug fix	WReaderGetIdentifier do not set pnIdentifierActualLength to the identifier length when returning W_ERROR_BUFFER_TOO_SHORT This value must be set to the actual identifier length to allow user to allocate the corresponding buffer
WME-1449	Bug fix	Memory leak when a user/client is destroyed In user/driver and client/server port, a memory leak occurs when the user/client is destroyed
WME-1452	Bug fix	Crash when WP2PRecvFrom() is requested with a buffer length egual to zero Memory leak in WP2PRecvFrom() or WP2PSendTo() when an error occurs
WME-1464 WME-1462	Bug fix	An assertion occurred, when the NFC controller is in STANDBY mode, and a client application calls the following sequence: WNFCControllerSwitchStandbyMode(true); WNFCControllerSwitchStandbyMode(false);
WME-1422	Bug fix	The WReaderListenToCardDetection function registers a handler function to listen to card detection. The WReaderListenToCardDetection function returns W_ERROR_BAD_PARAMETER and the event registry handle is set to W_NULL_HANDLE when priority is no access. The WReaderListenToCardDetection function registers a handler function to listen to card detection. The WReaderListenToCardDetection function returns W_ERROR_BAD_PARAMETER and the event registry handle is set to W_NULL_HANDLE when the array of connection properties contains one or some invalid connection properties.
WME-1484	Bug fix	The test agent crashes in monolithic porting when the connection center is not connected.

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1489	Bug fix	Exchange APDUs fails after 32 K operations When running stress tests exchanging APDUs with a 7816-4 contactless card, the test stops after typically 32 K iterations.
WME-1503	Bug fix	In WType1ChipWrite, when bLockBlocks is set to W_TRUE, the data are not written in the tag
WME-1510	Bug fix	The structure TWBTPairingInfo returned by WHandoverGetBluetoothInfo is invalid The structure is not initialized to zero before being filled.
WME-1542	Bug fix	Virtual tag created during connection handover cannot be read by the Nexus (the virtual tag is destroyed prior Nexus have read the content of the tag)
WME-1514	Bug fix	Cannot format a blank Mifare ULC after call to WMifareULForceULC ().
WME-1495	Bug fix	The W7816ExchangeAPDU function returns W_ERROR_BAD_PARAMETER while expected W_ERROR_BUFFER_TOO_SHORT when the receive APDU buffer is smaller than the data sent by the card.
WME-1501	Bug fix	W15SetTagSize function returns W_SUCCESS while expected W_ERROR_BAD_PARAMETER when sector size is set to 0
WME-1509	Bug fix	W_ERROR_BAD_STATE while expected W_SUCCESS by locking a Topaz card
WME-1516	Bug fix	The well-known service urn:nfc:sn:sdp is not recognized by the LLCP stack
WME-1500	Bug fix	W15Write function returns W_SUCCESS while expected W_ERROR_ITEM_LOCKED (error given by the card not properly parsed)
WME-1543	Bug fix	Several issues detected by Klocwork code analyser (possible memory leak, possible use of initialized values…)
WME-1533	Bug fix	No SYMM packet sent by P2P_TARGET when link deactivation has been requested
WME-1524	Bug fix	W15Read function returns W_ERROR_BAD_STATE while expected W_SUCCESS When W_ERROR_ITEM_LOCKED or W_ERROR_RF_COMMUNICATION occurs during a P15Write (or P15Read) operation, next P15Read or P15Write will fail with error W_ERROR_BAD_STATE
WME-1534	Bug fix	lock and lockable indicator are not updated after a lock operation
WME-1532	Bug fix	W_ERROR_BUFFER_TOO_SHORT when reading NDEF message from Virtual tags > 256 bytes
WME-1540	Bug fix	Missing manufacturer data in the Bluetooth pairing structure
WME-1541	Bug fix	RawMessage received when no RawListener - message still stored in RawMessageQueue When a RawMessage is received but there is no RawListener: static_PNFCControllerRawMessageEventReceived() places the RawMessage on the MessageQueue. (It does not call a listener callback since there isn't one.) When the next RawMessage is received (and this time there is a RawListener) then the message is placed on the queue and the listener callback is called with the newly received message length as a parameter. The callback function receives this length and will call WNFCControllerGetRawMessageData() expecting to read a message of this length. The user could create a buffer of this size to read the data. However the message ready to be dequeued is the first message and will necessarily be not be the expected length and the function would return with W_ERROR_BUFFER_TOO_SHORT

Copyright © 2012 Inside Secure

Id.	Type	Description
WME-1448	Bug fix	Default name for SE SWP slot should be "UICC xx"

Copyright © 2012 Inside Secure